PTO 08-5129

CC=JP DATE=19920706 KIND=A
PN=04188190

CHARACTER DRAWING DEVICE, CHARACTER OUTPUT COMPENSATION METHOD, AND CHARACTER OUTPUT DEVICE

[Mojibyogasochi to mojisyutsuryoku hoseihoho narabini mojisyutsuryoku sochi]

Shuichi Miura, et al.

UNITED STATES PATENT AND TRADEMARK OFFICE
Washington, D.C. May 2008

Translated by: FLS, Inc.

PUBLICATION COUNTRY	(19):	JP
DOCUMENT KIND	(12):	A
	(13):	PUBLISHED UNEXAMINED PATENT APPLICATION (Kokai)
PUBLICATION DATE	(43):	19920706 [WITHOUT GRANT]
PUBLICATION DATE	(45):	19920706 [WITH GRANT]
APPLICATION NUMBER	(21):	02-315951
APPLICATION DATE	(22):	19901122
PRIORITY DATE	(32):	
ADDITION TO	(61):	
INTERNATIONAL CLASSIFICATION	(51):	G09G 5/26; B41J 2/48
DOMESTIC CLASSIFICATION	(52):	
PRIORITY COUNTRY	(33):	
PRIORITY NUMBER	(31):	
PRIORITY DATE	(32):	
INVENTOR	(72):	MIURA, SHUICHI; YAMAZAKI, NAOMI; NAKAMURA, NAOKI; KAMIBAYASHI, HIROAKI
APPLICANT	(71):	HITACHI LTD.
TITLE	(54):	CHARACTER DRAWING DEVICE, CHARACTER OUTPUT COMPENSATION METHOD, AND CHARACTER OUTPUT DEVICE
FOREIGN TITLE	[54A]:	Mojibyogasochi to mojisyutsuryoku hoseihoho narabini mojisyutsuryoku sochi

Specification /829[*]

1. Name of this invention

Character drawing device, character output compensation method, and character output device

2. Claim(s)

[Claim 1] A character drawing device consisting of a font memory part for storing the character outline coordinate data forming outline fonts, a data readout part which reads out character outline coordinate data from said font memory part corresponding to the character code transmitted from a host computer, a coordinate transforming part for transforming said character outline coordinate data into a specific character size by transforming the coordinates, and a filling part for filling the inside of the area formed by transformed character outline coordinate data; wherein said character drawing device is equipped with a line width correction part comprising

a closed graphic separation part which separates the character outline coordinate data transformed by said coordinate transformation part for each closed graphic form,

an integer space calculation part which extracts the y coordinate of horizontal line outline coordinate and the x coordinate of vertical line outline coordinate from the character outline coordinate data separated by said closed graphic separation part,

[*] Numbers in the margin indicate pagination in the foreign text.

acquires the respective spaces between the adjacent coordinate values for the above-mentioned extracted y coordinate of the horizontal line outline coordinate and the x coordinate of the vertical line outline coordinate, and makes said spaces into respective integer values,

a whole length correction part which totals the spaces which were made into integers by said integer space calculation part and corrects at least one space of the areas having spaces transformed into integers when a gap is created between said total value and the whole length of the character in order to make said total value coincide with the whole length; and

a coordinate point correction part which recalculates the above-mentioned y coordinate of the horizontal line outline coordinate and the x coordinate of the vertical line outline coordinate by adding the space corrected by said whole length correction part, computes the positions relative to the outline coordinates of horizontal and vertical lines prior to recalculation for each coordinate point of the above-mentioned character outline coordinate data separated by said closed graphic form separation part, and corrects the positions by linear mapping the relative position of the recalculated horizontal line and vertical line outline coordinates.

[Claim 2] The character drawing device according to Claim 1, wherein the whole length correction part sequentially corrects /830
the width one dot at a time in the descending order of the white area widths.

[Claim 3] The character drawing device according to Claim 2, wherein in the case of plural white areas having the largest width, the whole length correction part sequentially corrects the area dot by dot from the center area when the difference between said total space value and said whole length of a character is an odd number, but from the areas at both ends when said difference is an even number.

[Claim 4] The character drawing device according to Claim 1, wherein the whole length correction part sequentially corrects the width dot by dot from the black area having the greatest width.

[Claim 5] The character drawing device according to Claim 4, wherein in the case of plural areas having the largest width, the whole length correction part sequentially corrects the area dot by dot from the center area when the difference of said total space value and said whole length of a character is an odd number, but from the areas at both ends when said difference is an even number.

[Claim 6] A data processing device equipped with a character drawing device according to one of Claims 1 - 5, a host computer which transmits a printing command to said character drawing device, and an output device which displays a character image created by said character drawing device.

[Claim 7] The data processing device according to Claim 6, wherein the output device is a laser printer, a graphic display terminal, or liquid crystal display terminal.

[Claim 8] The character drawing device according to Claim 3, wherein the character outline coordinate data stored in the font memory part provides a characteristic that two horizontal lines or vertical lines having the same line width have the same line width after printing.

[Claim 9] The character drawing device according to Claim 5, wherein the character outline coordinate data stored in the font memory part provides a characteristic that two white areas having the identical width among the white areas surrounded by two horizontal lines or vertical lines in a single closed graphic form do not change their widths after printing.

[Claim 10] A character output correction method which acquires the y coordinate value of each horizontal line outline coordinate configuring the outputting object character transformed into an arbitrary size from the character coordinate data of the outline font of said character, calculates the space between two adjacent y coordinate values, and sets the difference between the integer value obtained by arranging the total of each space into an integer and the value obtained by totaling each integer designating each space as the adjustment amount in the vertical direction at the time of outputting said character.

[Claim 11] A character output correction method which acquires the x coordinate value of each vertical line outline coordinate configuring the outputting object character transformed into an arbitrary size from the character coordinate data of the outline font of said character, calculates the space between two adjacent x coordinate values, and sets the difference between the integer value obtained by arranging the total of each space into an integer and the value obtained by totaling each integer designating each space as the adjustment amount in the horizontal direction at the time of outputting said character.

[Claim 12] A character output correction method which acquires the y coordinate value of each horizontal line outline coordinate configuring the output object character transformed into an arbitrary size from the character coordinate data of the outline font of said character, calculates the space between two adjacent y coordinate values, acquires the difference between the integer obtained by making the total of each space an integer and the value prepared by making each space an integer and totaled, and corrects the spaces indicated as non-character areas among the above-mentioned spaces one dot at a time in the descending order of the space sizes in the direction making said difference smaller.

[Claim 13] A character output correction method which acquires the x coordinate value of each vertical line outline coordinate configuring the output object character transformed into an arbitrary size from the character coordinate data of the outline font of said character, calculates the space between two adjacent y coordinate values, acquires the difference between the integer of the total of each space and the value obtained by making each space an integer and totaled, and corrects the spaces indicated as non-character areas among the above-mentioned spaces one dot at a time in the descending order of the space sizes in the direction making said difference smaller.

[Claim 14] A character output device which outputs characters using outline fonts, wherein said device is equipped with a means of outputting a character corrected in the vertical and horizontal directions by the respective methods according to Claims 12 and 13.

[Claim 15] A character output device which outputs characters using outline fonts, wherein said device is equipped with a means of calculating the correction amounts in the vertical and horizontal directions by performing the respective methods according to Claims 10 and 11. /831

3. Detailed explanation of this invention

[Industrial field]

This invention pertains to a display device for laser beam printers, displays units, etc. and is particularly associated with a character drawing/outputting device suitable for outputting high quality characters using outline fonts.

[Prior art]

Outline fonts are one of formats for storing character data in a display device, such as a laser printer, display unit, etc. With this outline font format, the shapes of characters are stored as an array of outline coordinates of the character. To print this outline font, each outline coordinate value is converted into coordinates at first so as to obtain the desired printing character size, and then the inside of the converted outline coordinates is filled in. Although the outline font has the characteristic of flexible enlargement, contraction, rotation, deformation, etc., when printing a character of 64 x 64 dots or smaller, the widths of vertical and horizontal lines of a character become uneven due to a quantization error to lower the display quality of the character.

As a conventional technique of correcting this line width, the widths of the vertical lines and horizontal lines are stored in a table beforehand, and when printing the character, the line width stored in the table is referenced using one of two outlines forming vertical lines and horizontal lines as a standard line so as to determine the outline coordinate of the other line for correcting the line width in order to realize high quality printing.

Note that there is a method disclosed in JP-A (Tokkai) H2-81657 as prior art.

[Problems to be solved by this invention]

However, the above-mentioned prior art must store line width information in a table for each character size beforehand. Therefore, this method restricts the printable character size based on the existence/non-existence of line width information. In addition, although the above-mentioned prior art can correct the line widths of horizontal and vertical lines, no consideration is made for correcting the white area surrounded by two horizontal lines or two vertical lines.

[Purpose of this invention]

The purpose of this invention is to provide a character drawing device or the like capable of correcting the line width of arbitrary character size for outline fonts not provided with additional information for correcting the line widths.

Another purpose of this invention is to provide a high quality character drawing device or the like which not only corrects the widths of horizontal and vertical lines but provides the consideration for correcting the white area surrounded by two horizontal lines or two vertical lines.

[Method to solve the problems]

The above-mentioned purposes can be achieved by extracting the outline coordinates of a horizontal line and a vertical line from outline coordinates generated after the coordinate conversion, making the spaces between the outline coordinates into respective integers, and recalculating each outline coordinate based on the obtained integers expressing the spaces.

Incidentally, the whole length expands or contracts by making the spaces between respective outline coordinates into integers. However, the whole length is corrected by fixing one side of a black area and a white area in uniform widths and correcting the other side by weighing heavily on its symmetry.

[Operation]

An integer-space calculation part automatically extracts the outline coordinate values of horizontal line and vertical line from character outline coordinate data and makes each space into a respective integer. As a result, even if the line width correction information for the arbitrary character sizes is not prepared beforehand, the spaces between the outline coordinates can remain the same before and after making the spaces into integers.

Moreover, since the whole length correction part provides correction by weighing heavily on the symmetrical property, high quality correction capable of maintaining the symmetrical property can be provided.

[Embodiment]

Hereafter, an embodiment of this invention will be explained by referring to Fig. 1 - Fig. 9.

Fig. 1 is a diagram illustrating the configuration of a character drawing device associated with an embodiment of this invention. When a host computer 1 sends a printing command to this character drawing device 2, the character drawing device 2 creates a character image with the reception of the above-mentioned printing command and transfers the data to a display device 3. The display device 3 is a laser printer, graphic display, or liquid crystal display device which prints or displays the above-mentioned character image.

In the character drawing device 2, a data read-out part 20 /832 decodes the font type information and the character code sent from the host computer 1 and reads out the specified character outline coordinate data to a work memory 25 from the font storage part 20. A coordinate transforming part 22 receives the coordinate information from the host computer 1 and transforms the character outline coordinate data readout to the work memory 25 specifically so as to obtain specific sizes. At this time, if the character outline coordinate data includes expressions of curves, such as a bijective curve, spline curve, arc, etc., the linear approximation process is performed by the coordinate transforming part 22. The character outline coordinate data having been processed by the transformation process is corrected by a line width correction part 23. A filling part 24 creates a character image by filling the inside surrounded by outlines based on the corrected character outline coordinate data and transfers the image to the display device 3.

In the line width correction part 23, a closed graphic form separation part 230 separates the character outline coordinate data having been transformed by the coordinate transformation part 22 into a group of coordinate data arrays forming a closed graphic form, cuts out one group at a time, and specifies the range to an integer space extraction part 231. Thereafter, the line width correction part 23 sequentially performs the succeeding processes on each group divided by the closed graphic form separation part 230.

Fig. 2 shows the operation of integer space extraction part 231. In this figure, the process performed to the y coordinate of horizontal line outline coordinate of character “日” is used as an example for explaining the operation of said part 231. Although the following will only explain the process performed to the y coordinate of the horizontal line outline coordinate according to Fig. 2, the same process is performed to the x coordinate of the vertical line outline coordinate as well.

After the y coordinates (y0 - y6) of the horizontal line outline coordinates are extracted from the coordinate data groups specified by the closed graphic form separation part 230, the largest value and smallest value of y coordinates in the above-mentioned coordinate data groups are added to the y coordinate of said horizontal line outline coordinate. Then, after the values are sorted, and the duplicate coordinate values are deleted, the results are stored in a horizontal line table 1000.

Next, the differences (dy0 - dy5) of two adjacent coordinate values in the above-mentioned horizontal line table are stored to a horizontal line space table 1001. Furthermore, each element of horizontal line space table 1001 is made into an integer, and the results (idy0 - idy5) are stored in a horizontal line space integer table 1002.

In addition, after the total (yt) of elements in the horizontal line space table 1001 is acquired and stored in a whole length table 1003, a whole integer length (iyt) provided by making said total length (yt) into an integer is stored in a whole integer length table 1005. Furthermore, the total (iys) of integers of horizontal line spaces (idy0 - idy5) stored in the horizontal line space integer table 1002 is acquired and stored in a horizontal line integer-space total table 1004.

Next, the difference (dyt) between the integer total (iys) of horizontal line spaces stored in a horizontal line integer-space total table 1004 and the whole integer length (iyt) stored in a whole integer length table 1005 is acquired and stored in a whole length error table 1006. The whole length error (dyt) stored in the whole length error table 1006 is used by the whole length correction part 232.

On the other hand, at the time of extracting the y coordinate of the above-mentioned horizontal line outline coordinate, each horizontal line outline coordinate is determined whether it is the upper side or lower side of the linear portion, and the obtained result is stored in a horizontal line attribute table 1007. Next, the area having its top side surrounded by the upper side and its bottom side surrounded by the lower side is determined as a black area, and the rest of the area is determined as a white area. Then, the result is stored in an area attribute table 1008. The information stored in the area attribute table 1008 is used by the whole length correction part 232.

The whole length correction part 232 first examines the whole length error (dyt) stored in the whole length error table 1006, and if the whole length error (dyt) is not 0, a correction process is performed. On the other hand, if it is 0, the next coordinate point correction part 233 is activated without performing any correction.

The whole length correction part 232 corrects the whole length by the performing the process shown in Fig. 3 when the whole length error (dyt) is not 0. First, at step 2000, the contents of the modifiable table 1009 are initialized to unmodifiable elements and modifiable elements corresponding to the attribute of the black area and white area in the area attribute table 1008. Next, at step 2001, the elements in the horizontal line space integer table 1002 whose corresponding contents in the modifiable table 1009 are modifiable are listed up. Hereafter, this will be called "modifiable object area".

Here, the absolute value of whole length error (dyt) is expressed as "adyt", and the result of dyt ÷ adyt is expressed as /833
"sdyt". At step 2002, if the number of said modifying object areas is equal to or smaller than adyt, sdyt is added to the width of every modifiable object area at step 2003. Then, if dyt is 0 or greater at

step 2004, the contents in the modifiable table 1009 corresponding to said modifying object areas is made unmodifiable at step 2005.

On the other hand, if the number of above-mentioned modifying object areas is equal or greater than adyt at step 2002, the whole length error is distributed in the modifying object areas. At this time, sdyt is sequentially added to the widths of modifying object areas starting from the center area at step 2007 if the above-mentioned adyt is an odd number at step 2006; if adyt is an even number at step 2006, however, sdyt is sequentially added to the modifying object areas from both end areas of the above-mentioned modifying object areas at step 2008.

After the process described above is performed, the number of the above-mentioned modifying object areas is subtracted from adyt. If adyt is resulted in 0 or less, the whole length correction process is ended to initiate the next coordinate point correction part 233; on the other hand, if adyt is greater than 0, step 2002 is performed.

Fig. 4 and Fig. 5 are diagrams illustrating the changes of the character figure pattern provided by the above-mentioned whole length correction part 232. Fig. 4 illustrates the case when the number of areas having the greatest value is less than adyt, and steps 2002, 2003, 2004, 2005, 2009, and 2010 are repeated three times. The black areas having the width of 1 do not change, while the width of the white area increases dot by dot sequentially in the descending order of the widths. Fig. 5 shows the case when plural areas having the greatest width exist for the number greater than adyt. The diagram shown in the left is the case when the correction amount adyt is an odd number, resulting in the operation of steps 2002, 2006, and 2007 in order to sequentially correct the width from the central white area. On the other hand, the diagram shown in the right is the case when the correction amount adyt is an even number, resulting in the operation of step 2002, 2006, and 2008 to correct the width sequentially from the white areas at both ends.

After the process of whole length correction part 232 ends as described above, the coordinate point correction part 233 corrects the above-mentioned character outline coordinate data. The coordinate point correction part 233 first determines the outline coordinate values of horizontal and vertical lines and then sequentially corrects each coordinate point by referencing the outline coordinate values of the above-mentioned horizontal and vertical lines.

Fig. 6 is a diagram illustrating the process of determining the outline coordinate value of the horizontal line. By adding each element in the horizontal line space integer table 1002, the corrected horizontal line outline coordinates (ny0 - ny6) are created and stored in the corrected horizontal line table 1010. That is, nyl is calculated by adding the integer space idy0 to the reference point coordinate ny0. Then, idyl is added to nyl to obtain ny2.

Thereafter, the process of adding nyi and idyi to obtain ny(i + 1) is repeated.

Once every corrected horizontal line (ny0 - ny6) is acquired, all the coordinate points are corrected by performing the following process on each coordinate point. First, the correcting coordinate value is compared with the elements in the horizontal line table 1000 to determine the area where said coordinate point belongs. Next, assuming that the coordinate point y exists between yi and y(i + 1) as shown in Fig. 7, the corrected coordinate point ny is obtained by the following formula:

$$\frac{y - yi}{y(i+1) - yi} = \frac{ny - nyi}{ny(i+1) - nyi}$$

The above-mentioned embodiment provides the effect of printing or displaying characters having uniform horizontal/vertical line widths without affecting the symmetric property.

The following explains another embodiment by referring to Fig. 7 and 8. According to this embodiment, the contents of modifiable table 1009 are initialized to modifiable components and unmodifiable components corresponding to the attributes of black areas and white areas in the area attribute table 1008.

Figs. 7 and 8 are diagrams illustrating the changes of character graphic pattern provided by the above-mentioned whole length correction part 232. Fig. 7 shows the case when the number of areas having the greatest width is less than ady, resulting in the operation of third steps 2002, 2003, 2004, 2005, 2009, and 2010 repeated three times. The white area having a width of 1 does not change, whereas the width of the black area sequentially increases dot by dot in the descending order of the widths. Fig. 9 shows the case when plural areas having the greatest width exist for the number which is larger than adyt. The left diagram in the figure illustrates the case when the correction amount adyt is an odd number, resulting in the operation of steps 2002, 2006, and 2007 so as to sequentially correct the width from the black area positioned at the center. On the other hand, the right diagram in the figure illustrates the case when the correction amount adyt is an even number, resulting in the operation of steps 2002, 2006, and 2008 to /834 sequentially correct the width from the black areas at both ends.

The above-mentioned embodiment provides the effect of printing or displaying characters having uniform horizontal/vertical line spaces without affecting the symmetric property.

[Effects of this Invention]

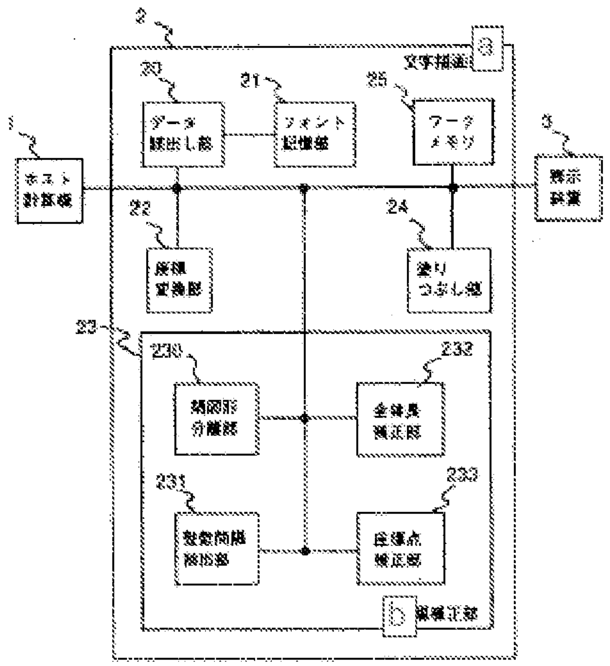
Based on the present invention, the line width of an outline font not provided with the additional information for correcting the line width can be corrected for an arbitrary character size. Moreover, the method of this invention can correct not only the horizontal and vertical line widths but also the white area surrounded by two horizontal or vertical lines, thereby being able to offer high quality characters.

4. Simple explanation of the figures

Fig. 1 is a character drawing device based on an embodiment of this invention. Fig. 2 is an explanatory diagram for explaining the process method of an integer space extraction part. Fig. 3 is a flowchart of the processing method of whole length correction part. Fig. 4 is an explanatory diagram of the first processing result of the overall correction part. Fig. 5 is an explanatory diagram of the first process of a coordinate point correction part. Fig. 6 is an explanatory diagram of the first process of a coordinate point correction part. Fig. 7 is an explanatory diagram of the second process of a coordinate point correction part. Fig. 8 is an explanatory diagram of the first process result of a whole length correction part in the second embodiment. Fig. 9 is an explanatory diagram for explaining the second process results of a whole length correction part of the second embodiment.

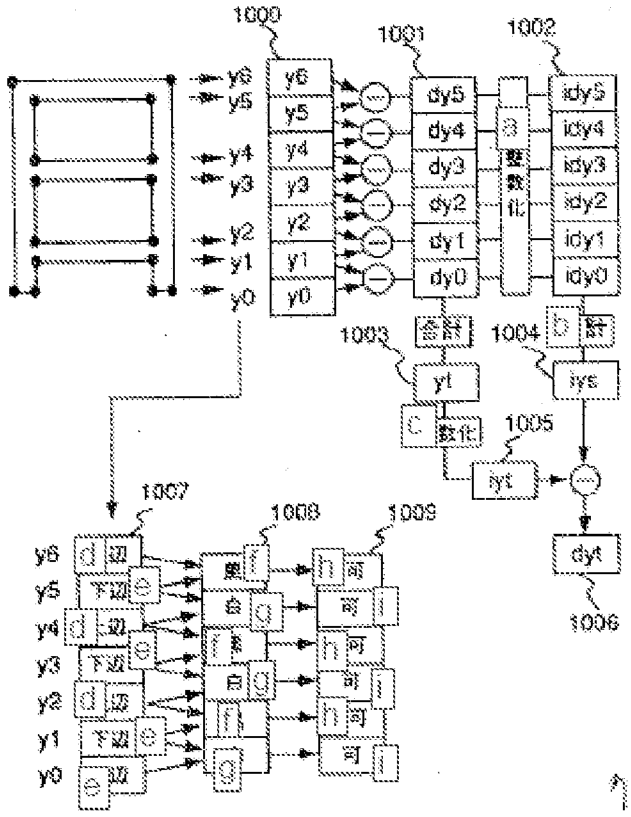
1...Host computer; 2..Character drawing device; 3...Display device; 20...Font memory part; 21...Data read out part; 22...Coordinate transformation part; 23...Line width correction part; 24...Filling part; 230...Closed graphic separation part; 231...Integer space extraction part; 232...Whole length correction part; 233...Coordinate point correction part

Figure 1



Key:
a) Character drawing device;
b) Line width correction part;
1...Host computer;
2...Character drawing device;
3...Display device;
20...Font memory part;
21...Data read out part;
22...Coordinate transformation part;
23...Line width correction part;
24...Filling part;
230...Closed graphic separation part;
231...Integer space extraction part;
232...Whole length correction part;
233...Coordinate point correction part

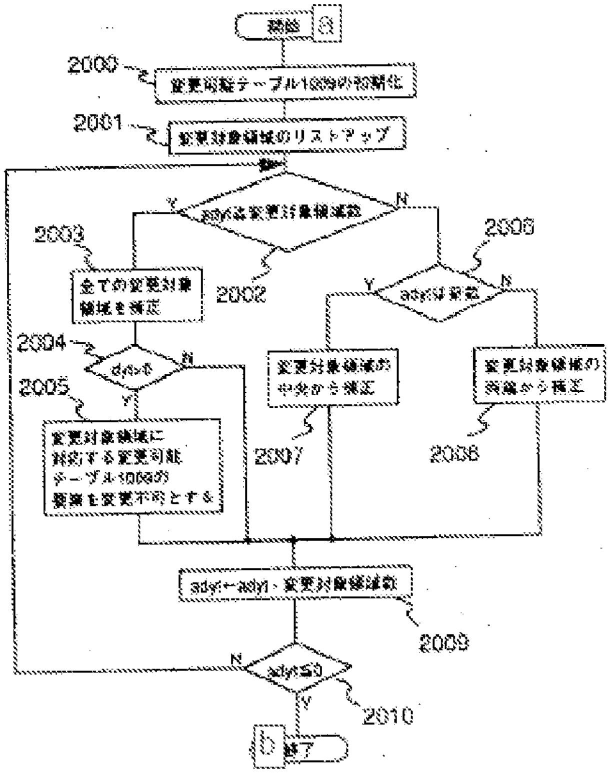
Figure 2



Key:
a) Process of transforming into integer;
b) Total;
c) Process of transforming into integer;
d) Upper side;
e) Lower side;
f) Black;
g) White;
h) Impossible;
i) Possible

Figure 3

/835


a
開始
2000
変更可能テーブル1009の初期化
2001
変更対象領域のリストアップ
adyl≧変更対象領域数
Y
N
2002
2003
全ての変更対象領域を補正
2004
2005
Y
N
変更対象領域に対応する変更可能テーブル1009の要素を変更不可とする
2006
adylは奇数
Y
N
変更対象領域の中央から補正
変更対象領域の両端から補正
2007
2008
adyl←adyl・変更対象領域数
2009
N
adyl≦0
Y
2010
b
終了

Key:
a) Start;
b) End;
2000...Initialization of modifiable table 1009;
2001...List up modifiable object areas;
2002...adyl ≥ Number of modifiable object area?;
2003...Correct all the modifiable object areas;
2005...Make the elements in modifiable tale 1009 unmodifiable corresponding to modifiable object areas;
2006...Is adyl an odd number?;
2007...Correct from the center of modifying object areas;
2008...Correct from both ends of modifying object areas;
2009...adyl - adyl·number of modifying object areas

Figure 4

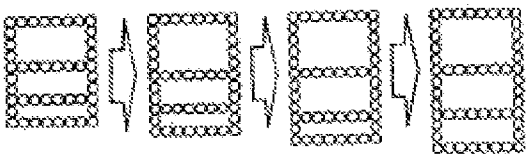
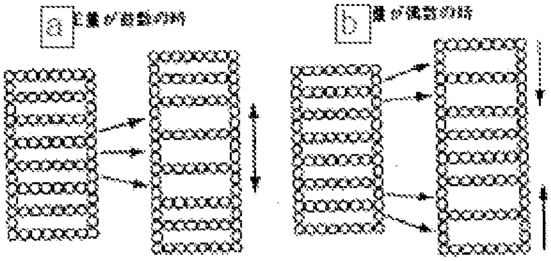


Figure 5



Key:
a) When correction amount is an odd number;
b) When correction amount is an even number

Figure 6

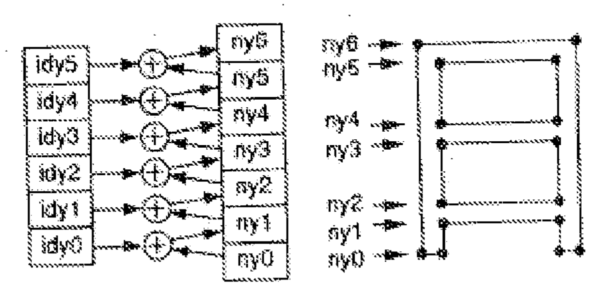


Figure 8

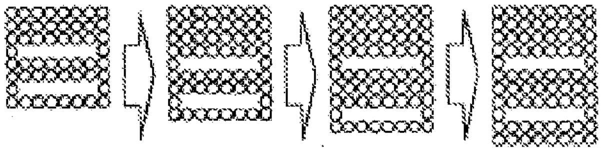


Figure 7

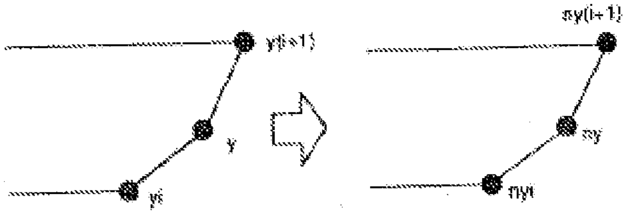
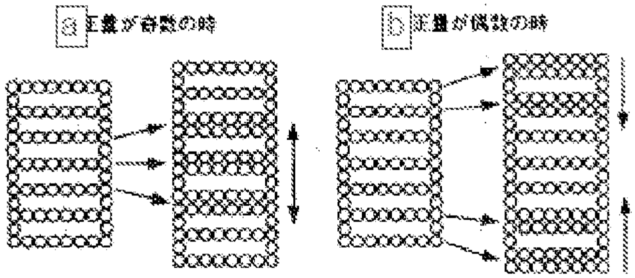


Figure 9



Key:
a) When correction amount is an odd number;
b) When correction amount is an even number